

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re U.S. Patent Application)
)
Applicant: Satoshi, Sakamoto)
)
Serial No.)
)
Filed: January 23, 2001)
)
For: INFORMATION PROCESSING)
APPARATUS EXECUTING PROCESSING)
CORRESPONDING TO NEW THREAD)
BY REUSING ARRANGEMENT FOR)
PREVIOUS THREAD)
)
Art Unit:)

*I hereby certify that this paper is being deposited
with the United States Postal Service as EXPRESS
mail in an envelope addressed to: Assistant
Commissioner for Patents, Washington, D.C. 20231,
on January 22, 2001.*

Express Label No.: EL 769181108 05

Signature: JR-

JC878 U.S. PTO
09/767991

CLAIM FOR PRIORITY

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Applicant claims foreign priority benefits under 35 U.S.C. § 119 on the basis
of the foreign application identified below:

Japanese Patent Application No. 2000-116073, filed April 18, 2000.

A certified copy of the priority document is enclosed.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By: 

Patrick G. Burns
Reg. No. 29,367

January 22, 2001
300 South Wacker Drive
Suite 2500
Chicago, IL 60606
(312) 360-0080
Customer Number: 24978

日 本 国 特 許 庁

PATENT OFFICE
JAPANESE GOVERNMENT

UC878 U.S. PTO
09/767991
01/23/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日

Date of Application:

2000年 4月18日

出 願 番 号

Application Number:

特願2000-116073

出 願 人

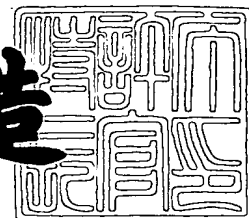
Applicant (s):

富士通株式会社

2000年10月27日

特許庁長官
Commissioner,
Patent Office

及 川 耕 造



出証番号 出証特2000-3089790

【書類名】 特許願

【整理番号】 9951974

【提出日】 平成12年 4月18日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 09/46

【発明の名称】 情報処理装置および情報処理方法

【請求項の数】 4

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 坂本 智

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100092152

 【弁理士】

 【氏名又は名称】 服部 毅巖

 【電話番号】 0426-45-6644

【手数料の表示】

 【予納台帳番号】 009874

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

 【包括委任状番号】 9705176

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 情報処理装置および情報処理方法

【特許請求の範囲】

【請求項 1】 アプリケーションプログラムからの要求に応じて処理の最小単位であるスレッドを生成して実行する情報処理装置において、

前記アプリケーションプログラムからのスレッド生成要求を受け付けるスレッド生成要求受け付け手段と、

前記スレッド生成要求受け付け手段によって受け付けられたスレッド生成要求に応じてスレッドを生成するスレッド生成手段と、

前記スレッド生成手段によって生成されたスレッドを実行する第 1 のスレッド実行手段と、

前記スレッド生成手段によるスレッドの生成に失敗したスレッド生成要求を保持するスレッド生成要求保持手段と、

前記スレッド生成要求保持手段に保持されているスレッド生成要求に対応する処理を実行する第 2 のスレッド実行手段と、

を有することを特徴とする情報処理装置。

【請求項 2】 前記第 2 のスレッド実行手段は、前記第 1 のスレッド実行手段によるスレッドの実行が終了した場合には、前記スレッド生成手段により新たなスレッドを生成することなく、使用済みのスレッドを再利用してスレッド生成要求に対応する処理を実行することを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 アプリケーションプログラムからの要求に応じて処理の最小単位であるスレッドを生成して実行する情報処理方法において、

前記アプリケーションプログラムからのスレッド生成要求を受け付けるスレッド生成要求受け付けステップと、

前記スレッド生成要求受け付けステップによって受け付けられたスレッド生成要求に応じてスレッドを生成するスレッドステップと、

前記スレッド生成ステップによって生成されたスレッドを実行する第 1 のスレッド実行ステップと、

前記スレッド生成ステップによるスレッドの生成に失敗したスレッド生成要求

を保持するスレッド生成要求保持ステップと、

前記スレッド生成要求保持ステップに保持されているスレッド生成要求に対応する処理を実行する第 2 のスレッド実行ステップと、

を有することを特徴とする情報処理方法。

【請求項 4】 アプリケーションプログラムからの要求に応じて処理の最小単位であるスレッドを生成して実行する処理をコンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体において、

コンピュータを、

前記アプリケーションプログラムからのスレッド生成要求を受け付けるスレッド生成要求受け付け手段、

前記スレッド生成要求受け付け手段によって受け付けられたスレッド生成要求に応じてスレッドを生成するスレッド生成手段、

前記スレッド生成手段によって生成されたスレッドを実行する第 1 のスレッド実行手段、

前記スレッド生成手段によるスレッドの生成に失敗したスレッド生成要求を保持するスレッド生成要求保持手段、

前記スレッド生成要求保持手段に保持されているスレッド生成要求に対応する処理を実行する第 2 のスレッド実行手段、

として機能させるプログラムを記録したコンピュータ読み取り可能な記録媒体

。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は情報処理装置に関し、特に、アプリケーションプログラムからの要求に応じて処理の最小単位であるスレッドを生成して実行する情報処理装置に関する。

【0002】

【従来の技術】

従来において、マルチスレッド OS (Operating System) と呼ばれるシステム

では、プログラム（プロセス）の中で、複数のスレッドを実行することが可能とされている。同じプロセスの中のスレッド同士は、メモリ資源等を共有することができるので、スレッド間のデータ転送は比較的容易であるという特徴がある。

【0003】

図10は、マルチスレッドOSを採用した情報処理装置の一例を示す図である。この図において、アプリケーションプログラム10は、例えば、記憶装置に記憶されており、必要に応じて起動され、実行される。

【0004】

スレッド処理部20は、アプリケーションプログラムからスレッドの生成要求がなされた場合には、スレッドを生成して処理する。OS30はオペレーティングシステムであり、スレッドの生成要求に応じてスレッドを生成する。

【0005】

スレッド管理処理部20aは、アプリケーションプログラム10からスレッド生成要求がなされた場合にはこれを入力し、スレッドの生成要求をOS30に対して行う。そして、スレッドの生成に成功した場合には、スレッド管理部20bに通知し、それ以外の場合にはスレッドの生成に失敗したとして、要求を行ったアプリケーションプログラム10に対してエラーを通知する。

【0006】

スレッド管理部20bは、スレッドの生成が成功した場合には、実行中スレッド管理部20cに対して新たなスレッドの管理を委託する。

実行中スレッド管理部20cは、スレッド実行部20d～20fによるスレッドの実行を管理する。

【0007】

スレッド実行部20d～20fは、新たなスレッドが生成された場合に生成され、生成されたスレッドを実行する。また、スレッドの生成が終了した場合には消滅する。

【0008】

次に、以上の従来例の動作について説明する。

いま、アプリケーションプログラム10からスレッドの生成要求があったとす

ると、スレッド管理処理部 20a がこれを受け、OS 30 に対してスレッドの生成要求を行う。その結果、スレッドの生成に成功した場合には、新たなスレッド実行部を生成するとともに、スレッド管理部 20b に対してスレッドの処理を委託する。

【0009】

スレッド管理部 20b は、実行中スレッド管理部 20c に対して新たに生成されたスレッドに関する情報を供与し、その実行を管理させる。その結果、実行中スレッド管理部 20c は、新たに生成されたスレッド実行部に対してスレッドの実行を割り当てる。

【0010】

一方、OS 30 がスレッドの生成に失敗した場合には、スレッド管理処理部 20a に対してスレッドの生成に失敗した旨が通知される。すると、スレッド管理処理部 20a は、要求を行ったアプリケーションプログラム 10 に対してエラーを通知する。なお、エラー通知を受けたアプリケーションプログラム 10 は、例えば、所定の時間だけ待機した後、スレッドの生成要求を再度行う。

【0011】

【発明が解決しようとする課題】

ところで、アプリケーションプログラム 10 がエラー通知を受けた場合には、前述のように再度の要求を行うことになる。しかしながら、アプリケーションプログラム 10 側では、スレッドの生成が可能になるタイミングが分からないため、例えば、頻繁に生成要求を行う必要があった。そのためスレッドの生成要求が頻発してシステム全体の処理が遅延される場合があるという問題点があった。

【0012】

また、スレッドの要求が失敗した場合には、別の機会にスレッド管理処理部 20a がスレッドの生成要求を OS 30 に対して再度行う必要が生ずるが、OS 30 がスレッドを生成するには多大な時間を要するため、このような処理が増加した場合には、前述の場合と同様にシステム全体の処理が遅延されるという問題点があった。

【0013】

本発明は、以上のような点に鑑みてなされたものであり、マルチスレッドOSを採用した情報処理装置の実行速度を向上することを目的とする。

【0014】

【課題を解決するための手段】

本発明では上記課題を解決するために、図1に示す、アプリケーションプログラム1からの要求に応じて処理の最小単位であるスレッドを生成して実行する情報処理装置において、前記アプリケーションプログラム1からのスレッド生成要求を受け付けるスレッド生成要求受け付け手段2aと、前記スレッド生成要求受け付け手段2aによって受け付けられたスレッド生成要求に応じてスレッドを生成するスレッド生成手段2bと、前記スレッド生成手段2bによって生成されたスレッドを実行する第1のスレッド実行手段2cと、前記スレッド生成手段2bによるスレッドの生成に失敗したスレッド生成要求を保持するスレッド生成要求保持手段2dと、前記スレッド生成要求保持手段2dに保持されているスレッド生成要求に対応する処理を実行する第2のスレッド実行手段2eと、を有することを特徴とする情報処理装置が提供される。

【0015】

ここで、スレッド生成要求受け付け手段2aは、アプリケーションプログラム1からのスレッド生成要求を受け付ける。スレッド生成手段2bは、スレッド生成要求受け付け手段2aによって受け付けられたスレッド生成要求に応じてスレッドを生成する。第1のスレッド実行手段2cは、スレッド生成手段2bによって生成されたスレッドを実行する。スレッド生成要求保持手段2dは、スレッド生成手段2bによるスレッドの生成に失敗したスレッド生成要求を保持する。第2のスレッド実行手段2eは、スレッド生成要求保持手段2dに保持されているスレッド生成要求に対応する処理を実行する。

【0016】

【発明の実施の形態】

以下、本発明の実施の形態を図面を参照して説明する。

図1は、本発明の動作原理を説明する原理図である。この図において、アプリケーションプログラム1は、スレッド処理部2に対してスレッドの生成要求を行

う。

【0017】

スレッド処理部2は、スレッド生成要求受け付け手段2a、スレッド生成手段2b、第1のスレッド実行手段2c、スレッド生成要求保持手段2d、および、第2のスレッド実行手段2eによって構成されており、アプリケーションプログラム1からのスレッドの生成要求に応じてスレッドを生成し、そのスレッドに係る処理を実行する。

【0018】

ここで、スレッド生成要求手段受け付け2aは、アプリケーションプログラム1からのスレッド生成要求を受け付ける。

スレッド生成手段2bは、スレッド生成要求受け付け手段2aによって受け付けられたスレッド生成要求に応じてスレッドを生成する。

【0019】

第1のスレッド実行手段2cは、スレッド生成手段2bによって生成されたスレッドを実行する。

スレッド生成要求保持手段2dは、スレッド生成手段2bによるスレッドの生成に失敗したスレッド生成要求を保持する。

【0020】

第2のスレッド実行手段2eは、スレッド生成要求保持手段2dに保持されているスレッド生成要求に対応する処理を実行する。

次に、以上の原理図の動作について説明する。

【0021】

いま、アプリケーションプログラム1からスレッドの生成要求がなされたとすると、スレッド処理部2のスレッド生成要求受け付け手段2aがこれを受け付ける。

【0022】

スレッド生成要求受け付け手段2aは、受け付けたスレッド生成要求を、スレッド生成手段2bに供給する。

スレッド生成手段2bは、スレッド生成要求受け付け手段2aから供給された

スレッド生成要求に応じて新たなスレッドを生成する。そして、新たなスレッドの生成に成功した場合には、スレッドの実行要求を第 1 のスレッド実行手段 2 c に対して行い、失敗した場合にはスレッド生成要求保持手段 2 d に対してスレッドの生成要求を供給する。

【0023】

第 1 のスレッド実行手段 2 c は、スレッド生成手段 2 b によって生成されたスレッドを実行し、スレッドの実行が完了した場合にはその旨を第 2 のスレッド実行手段 2 e に通知する。

【0024】

いま、第 1 のスレッド実行手段 2 c ではスレッドが実行中である場合に、新たなスレッドの生成要求がアプリケーションプログラム 1 からなされ、スレッド生成手段 2 b がスレッドの生成に失敗したとすると、スレッド生成要求はスレッド生成要求保持手段 2 d に供給されることになる。

【0025】

スレッド生成要求保持手段 2 d は、スレッド生成要求をスレッド生成手段 2 b から受け取り、保持する。

そして、第 1 のスレッド実行手段 2 c によるスレッドの実行が終了した場合には、その旨が第 2 のスレッド実行手段 2 e に対して通知される。

【0026】

第 2 のスレッド実行手段 2 e は、第 1 のスレッド実行手段 2 c によるスレッドの実行が完了したことを認知し、スレッド生成要求保持手段 2 d が保持しているスレッド生成要求のうち、最先のものを読み出す。そして、第 2 のスレッド実行手段 2 e は、読み出したスレッド生成要求に対して、第 1 のスレッド実行手段 2 c が使用済みのスレッドを再利用して、要求された処理の実行を開始する。

【0027】

以上に示したように、本発明に係る情報処理装置によれば、スレッドの生成に失敗した場合には、スレッド生成要求をスレッド生成要求保持手段 2 d に保持し、既存のスレッドの処理が終了した場合に読み出して処理するようにしたので、アプリケーションプログラムが再度のスレッド生成要求を行う必要がなくなり、

システム全体の処理速度を向上させることが可能となる。

【0028】

また、スレッド生成要求保持手段2dに保持されているスレッド生成要求に対しては、第1のスレッド実行手段2cによって実行が完了したスレッドを再利用して処理を実行するようにしたので、OSに対してスレッドの生成要求を再度行う必要がなくなるので、システム全体の処理速度を向上させることが可能となる。

【0029】

なお、以上の原理図においては、第1のスレッド実行手段2cと第2のスレッド実行手段2eとを別々の構成としたが、これらを統合して1つにしてもよい。

次に、図2を参照して、本発明の実施の形態の構成例について説明する。

【0030】

この図において、アプリケーションプログラム10は、例えば、記憶装置に記憶されており、必要に応じて起動されて実行される。スレッド処理部20は、アプリケーションプログラム10からスレッドの生成要求がなされた場合には、スレッドを生成して該当する処理を実行する。OS30はオペレーティングシステムであり、スレッドの生成要求に応じてスレッドを生成する。

【0031】

スレッド管理処理部20aは、アプリケーションプログラム10からスレッド生成要求がなされた場合にはこれを入力し、OS30に対して供給する。そして、OS30がスレッドの生成に成功した場合には、(1)アプリケーションプログラム10から対象となる処理に関する情報を取得し、(2)後述するスレッド制御表(図3参照)を生成し、(3)スレッド実行部を生成し、(4)スレッド管理部20bを介して実行中スレッド管理部20cに対してスレッドの管理(処理の実行要求)を委託する。

【0032】

また、スレッドの生成に失敗した場合には、(1)アプリケーションプログラム10から対象となる処理に関する情報を取得し、(2)後述するスレッド制御表(図3参照)を生成し、(3)スレッド管理部20bを介して待機中スレッド

管理部 2 0 g に対してスレッドの管理を委託する。

【 0 0 3 3 】

スレッド管理部 2 0 b は、スレッド管理処理部 2 0 a から委託された処理の実行要求を、スレッドの生成に成功したものと失敗したものとに分けて管理する。即ち、スレッドの生成に成功したものは、実行中スレッド管理部 2 0 c に登録し、また、失敗したものは待機中スレッド管理部 2 0 g に登録する。

【 0 0 3 4 】

実行中スレッド管理部 2 0 c は、スレッド管理部 2 0 b によって登録された処理の実行要求を、受け付けた順に記憶して管理する。

待機中スレッド管理部 2 0 g は、スレッド管理部 2 0 b によって登録された処理の実行要求を、受け付けた順に記憶して管理する。

【 0 0 3 5 】

スレッド実行部 2 0 d ~ 2 0 f は、OS 3 0 がスレッドの生成に成功した場合に生成され、対応するスレッドの実行を管理する。また、スレッドの実行が終了した場合であって、待機中スレッド管理部 2 0 g に待機中の処理の実行要求が存在している場合にはこれを取得して実行し、存在しない場合には消滅する。

【 0 0 3 6 】

次に、図 3 ~ 図 6 を参照して、図 1 に示す実施の形態で実行される処理の一例について説明する。

図 3 は、1 つのスレッドが実行中である場合における、スレッド管理部 2 0 b 、実行中スレッド管理部 2 0 c 、および、待機中スレッド管理部 2 0 g の関係を示す図である。この図に示すように、スレッド管理部 2 0 b は、実行中スレッド管理部 2 0 c が関連付けられた実行中スレッドキュー 2 0 b - 1 と、待機中スレッド管理部 2 0 g が関連付けられた待機中スレッドキュー 2 0 b - 2 とを有している。

【 0 0 3 7 】

実行中スレッド管理部 2 0 c は、実行中のスレッドの処理内容を示すスレッド制御表が格納されているアドレスを管理している。この例では、実行中のスレッドは既述のように 1 つであるので、スレッド制御表 4 0 のみが関連付けられてい

る。なお、スレッド制御表 4 0 には、そのプログラムのエントリアドレス、パラメタアドレス、および、作業領域アドレス等が格納されている。

【 0 0 3 8 】

いま、図 3 に示す、1 つのスレッドが実行されている状態において、アプリケーションプログラム 1 0 から新たなスレッドの生成要求がなされたとなると、スレッド管理処理部 2 0 a は、これを受け付け、OS 3 0 に対してスレッドの生成要求を行う。

【 0 0 3 9 】

その結果、スレッドの生成が成功した場合には、スレッド管理処理部 2 0 a は、スレッドの生成に成功した旨をアプリケーションプログラム 1 0 に通知するとともに、対象となる処理に関する情報（エントリアドレス、パラメタアドレス、および、作業領域アドレス）をアプリケーションプログラム 1 0 から取得し、図 4 に示すスレッド制御表 4 1 を生成する。そして、スレッドの生成に成功した旨と、生成されたスレッド制御表 4 1 が格納されているアドレス（\$ 0 0 4 5 4 1 ）とをスレッド管理部 2 0 b に対して供給する。

【 0 0 4 0 】

スレッド管理部 2 0 b は、供給されたスレッド制御表 4 1 のアドレスを、実行中スレッド管理部 2 0 c に格納する。その結果、実行中スレッド管理部 2 0 c には、図 4 に示すように、現在実行中のスレッドに関するスレッド制御表 4 0 のアドレス \$ 0 0 1 0 2 2 と、新たに生成されたスレッドに関するスレッド制御表 4 1 のアドレス \$ 0 0 4 5 4 1 とが格納されることになる。

【 0 0 4 1 】

実行中スレッド管理部 2 0 c に対してアドレスが格納されると、その格納位置に対応したスレッド実行部が生成され、スレッドの処理が開始される。いまの例では、例えば、スレッド実行部 2 0 d が生成されてスレッドが実行される。

【 0 0 4 2 】

続いて、新たなスレッド生成要求がアプリケーションプログラム 1 0 からなされたとなると、前述の場合と同様の処理によりスレッドが生成され、図 5 に示すように、実行中スレッド管理部 2 0 c に対してスレッド制御表 4 2 のアドレス \$

0 0 A B 2 4 が格納され、対応する処理が実行されることになる。

【 0 0 4 3 】

続いて、新たなスレッド生成要求がアプリケーションプログラム 1 0 から更になされたとすると、スレッド管理処理部 2 0 a は O S 3 0 に対してスレッドの生成要求を行うが、スレッドは既に 3 つ生成されていることから、新たな生成は許可しない旨の通知が O S 3 0 からなされることになる。

【 0 0 4 4 】

その場合、スレッド管理処理部 2 0 a は、スレッドの生成には失敗したものの、アプリケーションプログラム 1 0 に対してはスレッドの生成に成功した旨の通知を行い、対象となる処理に関する情報を取得する。そして、取得した情報に基づいて、図 6 に示すスレッド制御表 4 3 を生成し、スレッド管理部 2 0 b に対してスレッド制御表 4 3 のアドレス \$ 0 0 F D 5 1 を供給して管理させる。

【 0 0 4 5 】

その結果、供給されたスレッド制御表 4 3 のアドレス \$ 0 0 F D 5 1 は待機中スレッド管理部 2 0 g に格納されることになる。

このような状態において、スレッド制御表 4 0 に係る処理が終了したとする。このとき、スレッド制御表 4 0 に係る処理がスレッド実行部 2 0 d によって実行されていたとすると、スレッド実行部 2 0 d は、待機中スレッド管理部 2 0 g を参照し、アドレスが格納されているか否かを判定する。いまの例では、アドレス \$ 0 0 D F 5 1 が存在していることから、スレッド実行部 2 0 d はこのアドレス \$ 0 0 D F 5 1 を取得し、実行中スレッド管理部 2 0 c の自己の領域（この例では最上部）に対して取得したアドレスを格納するとともに、該当するスレッド制御表を参照して要求されている処理を実行する。その結果、図 7 に示すように、実行中スレッド管理部 2 0 c に対してスレッド制御表 4 3 に対応するアドレスが格納され、スレッド実行部 2 0 d によってスレッドに係る処理が実行されることになる。

【 0 0 4 6 】

以上に説明したように、本発明の実施の形態によれば、スレッドの生成に失敗した場合には、待機中スレッド管理部 2 0 g に対してスレッド生成要求を格納し

、処理が可能となった場合に実行中スレッド管理部 2 0 c に移して実行するようにしたので、アプリケーションプログラム 1 0 に係る負担を軽減し、システム全体の処理速度を向上させることが可能となる。

【 0 0 4 7 】

また、実行中スレッド管理部 2 0 c によるスレッドの実行が終了した場合には、当該使用済みスレッドに対して待機中の要求に係る処理を割り当てて実行するようにしたので、スレッドを再生成することを回避し、その結果、システム全体としての処理速度を向上させることが可能となる。

【 0 0 4 8 】

最後に、以上の実施の形態において実行される処理の一例について説明する。

図 8 は、図 2 に示すスレッド管理処理部 2 0 a において実行される処理の一例を説明するフローチャートである。このフローチャートが開始されると、以下の処理が実行される。

〔 S 1 〕 スレッド管理処理部 2 0 a は、アプリケーションプログラム 1 0 からスレッドの生成要求があったか否かを判定し、あった場合にはステップ S 2 に進み、それ以外の場合にはステップ S 1 に戻って同様の処理を繰り返す。

〔 S 2 〕 スレッド管理処理部 2 0 a は、 O S 3 0 に対してスレッドの生成要求を行う。

〔 S 3 〕 スレッド管理処理部 2 0 a は、スレッドの生成に成功したか否かを判定し、成功した場合にはステップ S 4 に進み、それ以外の場合にはステップ S 5 に進む。

〔 S 4 〕 スレッド管理処理部 2 0 a は、スレッド管理部 2 0 b を介して、実行中スレッド管理部 2 0 c にスレッドに係る情報を登録する。

〔 S 5 〕 スレッド管理処理部 2 0 a は、スレッド管理部 2 0 b を介して、待機中スレッド管理部 2 0 g にスレッドに係る情報を登録する。

〔 S 6 〕 スレッド管理処理部 2 0 a は、処理を継続するか否かを判定し、継続する場合にはステップ S 1 に戻って同様の処理を繰り返し、それ以外の場合には処理を終了する。

【 0 0 4 9 】

次に、図 1 0 を参照して、図 2 に示すスレッド実行部 2 0 d ~ 2 0 f において実行される処理の一例について説明する。このフローチャートが開始されると、以下の処理が実行される。なお、以下ではスレッド実行部 2 0 d を例に挙げて説明する。

〔S 2 0〕スレッド実行部 2 0 d は、スレッドの実行処理を行う。

〔S 2 1〕スレッド実行部 2 0 d は、待機中スレッド管理部 2 0 g を参照し、待機中の処理要求が存在するか否かを判定する。その結果、待機中の処理要求が存在している場合にはステップ S 2 2 に進み、それ以外の場合には処理を終了する。

〔0 0 5 0〕

なお、処理が終了すると、当該スレッドは消滅することになる。

〔S 2 2〕スレッド実行部 2 0 d は、待機中スレッド管理部 2 0 g から、最先に受け付けられた処理要求を取得する。

〔S 2 3〕スレッド実行部 2 0 d は、実行中スレッド管理部 2 0 c に対して、取得した処理要求を供給する。

〔0 0 5 1〕

以上の処理によれば、図 2 に示す実施の形態を参照して説明した機能を実現することが可能となる。

なお、以上の実施の形態では、アプリケーションプログラムが 1 つだけの場合を例に挙げて説明したが、複数のアプリケーションプログラムが実行されている場合においても本発明を適用することが可能であることはいうまでもない。

〔0 0 5 2〕

また、以上の実施の形態では、スレッド制御部が 3 つの場合を例に挙げて説明したが、本発明はこのような場合のみに限定されるものではなく、それ以外の数の場合でも適用可能であることは勿論である。

〔0 0 5 3〕

最後に、上記の処理機能は、コンピュータによって実現することができる。その場合、情報処理装置が有すべき機能の処理内容は、コンピュータで読み取り可能な記録媒体に記録されたプログラムに記述されており、このプログラムをコン

コンピュータで実行することにより、上記処理がコンピュータで実現される。コンピュータで読み取り可能な記録媒体としては、磁気記録装置や半導体メモリ等がある。市場へ流通させる場合には、ＣＤ－ＲＯＭ(Compact Disk Read Only Memory)やフロッピーディスク等の可搬型記録媒体にプログラムを格納して流通させたり、ネットワークを介して接続されたコンピュータの記憶装置に格納しておき、ネットワークを通じて他のコンピュータに転送することもできる。コンピュータで実行する際には、コンピュータ内のハードディスク装置等にプログラムを格納しておき、メインメモリにロードして実行する。

【 0 0 5 4 】

【発明の効果】

以上説明したように本発明では、アプリケーションプログラムからの要求に応じて処理の最小単位であるスレッドを生成して実行する情報処理装置において、アプリケーションプログラムからのスレッド生成要求を受け付けるスレッド生成要求受け付け手段と、スレッド生成要求受け付け手段によって受け付けられたスレッド生成要求に応じてスレッドを生成するスレッド生成手段と、スレッド生成手段によって生成されたスレッドを実行する第１のスレッド実行手段と、スレッド生成手段によるスレッドの生成に失敗したスレッド生成要求を保持するスレッド生成要求保持手段と、スレッド生成要求保持手段に保持されているスレッド生成要求に対応する処理を実行する第２のスレッド実行手段と、を有するようにしたので、スレッドの生成要求が失敗した場合の処理を迅速に行うことが可能となるので、システム全体の処理速度を向上させることが可能となる。

【図面の簡単な説明】

【図 1】

本発明の動作原理を説明する原理図である。

【図 2】

本発明の実施の形態の構成例を示す図である。

【図 3】

スレッド管理部、実行中スレッド管理部、待機中スレッド管理部、および、スレッド制御表の対応関係を示す図である。

【図 4】

スレッド管理部、実行中スレッド管理部、待機中スレッド管理部、および、スレッド制御表の対応関係の他の一例を示す図である。

【図 5】

スレッド管理部、実行中スレッド管理部、待機中スレッド管理部、および、スレッド制御表の対応関係の他の一例を示す図である。

【図 6】

スレッド管理部、実行中スレッド管理部、待機中スレッド管理部、および、スレッド制御表の対応関係の他の一例を示す図である。

【図 7】

スレッド管理部、実行中スレッド管理部、待機中スレッド管理部、および、スレッド制御表の対応関係の他の一例を示す図である。

【図 8】

スレッド管理処理部において実行される処理の一例を説明するフローチャートである。

【図 9】

スレッド実行部において実行される処理の一例を説明するフローチャートである。

【図 1 0】

従来の情報処理装置の構成例を示す図である。

【符号の説明】

- 1 アプリケーションプログラム
- 2 スレッド処理部
 - 2 a スレッド生成要求受け付け手段
 - 2 b スレッド生成手段
 - 2 c 第 1 のスレッド実行手段
 - 2 d スレッド生成要求保持手段
 - 2 e 第 1 のスレッド実行手段
- 1 0 アプリケーションプログラム

20 スレッド処理部

20 a スレッド管理処理部

20 b スレッド管理部

20 c 実行中スレッド管理部

20 d ~ 20 f スレッド実行部

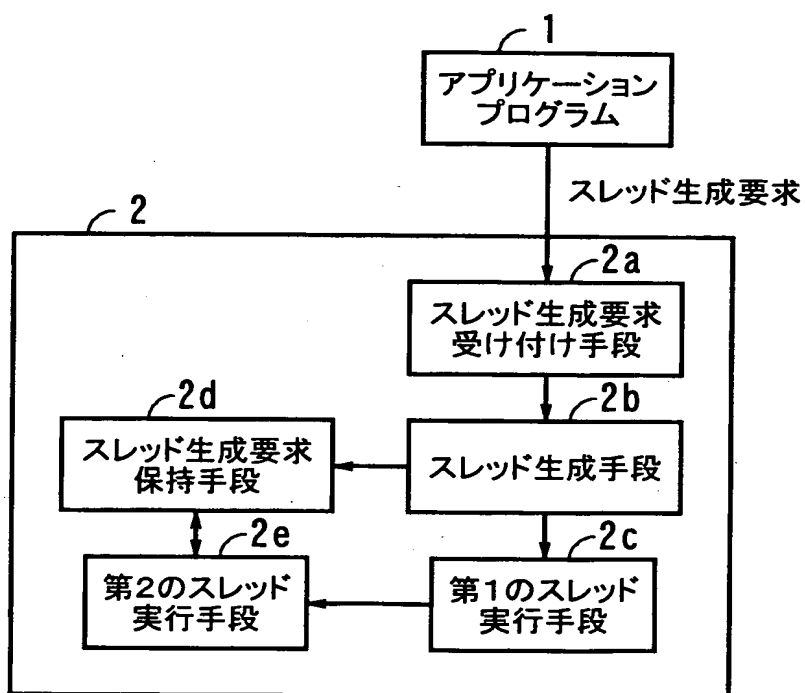
20 g 待機中スレッド管理部

30 OS

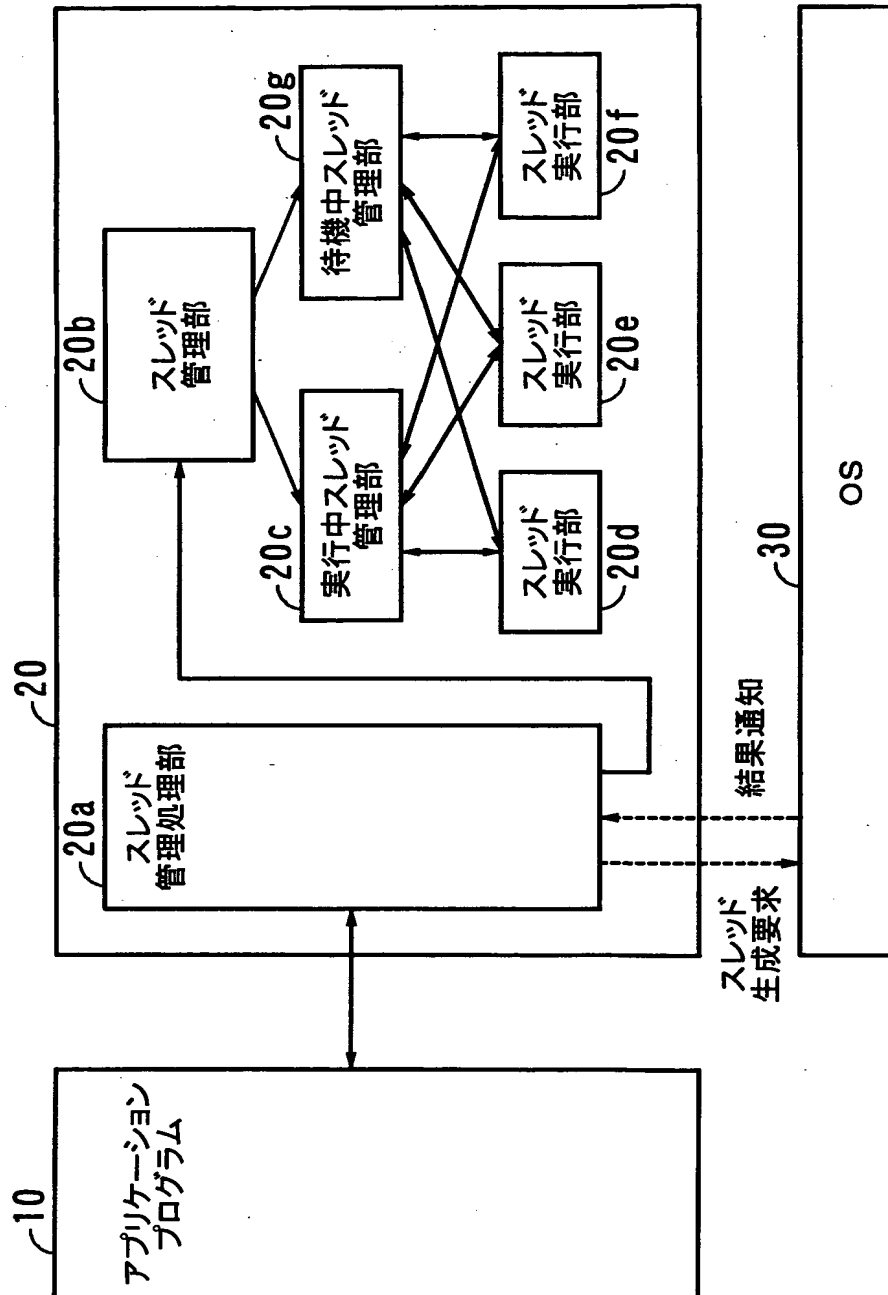
【書類名】

図面

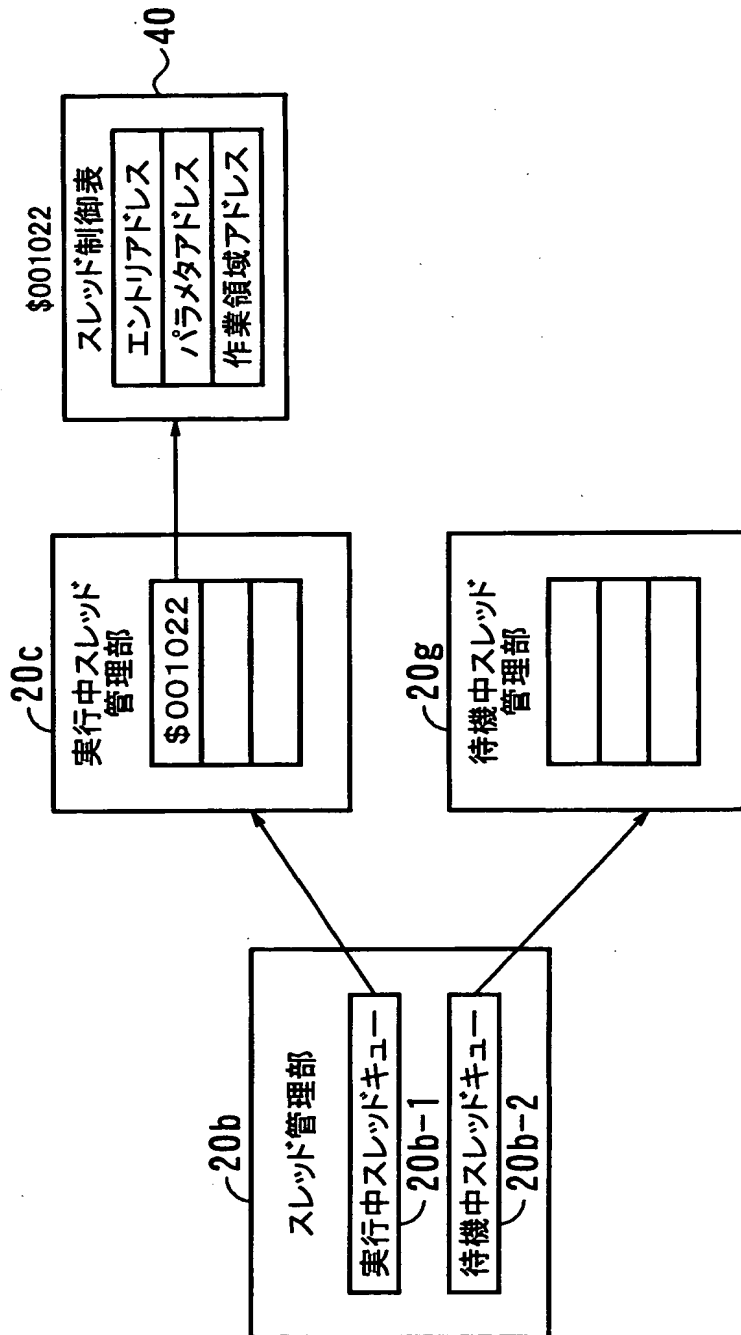
【図1】



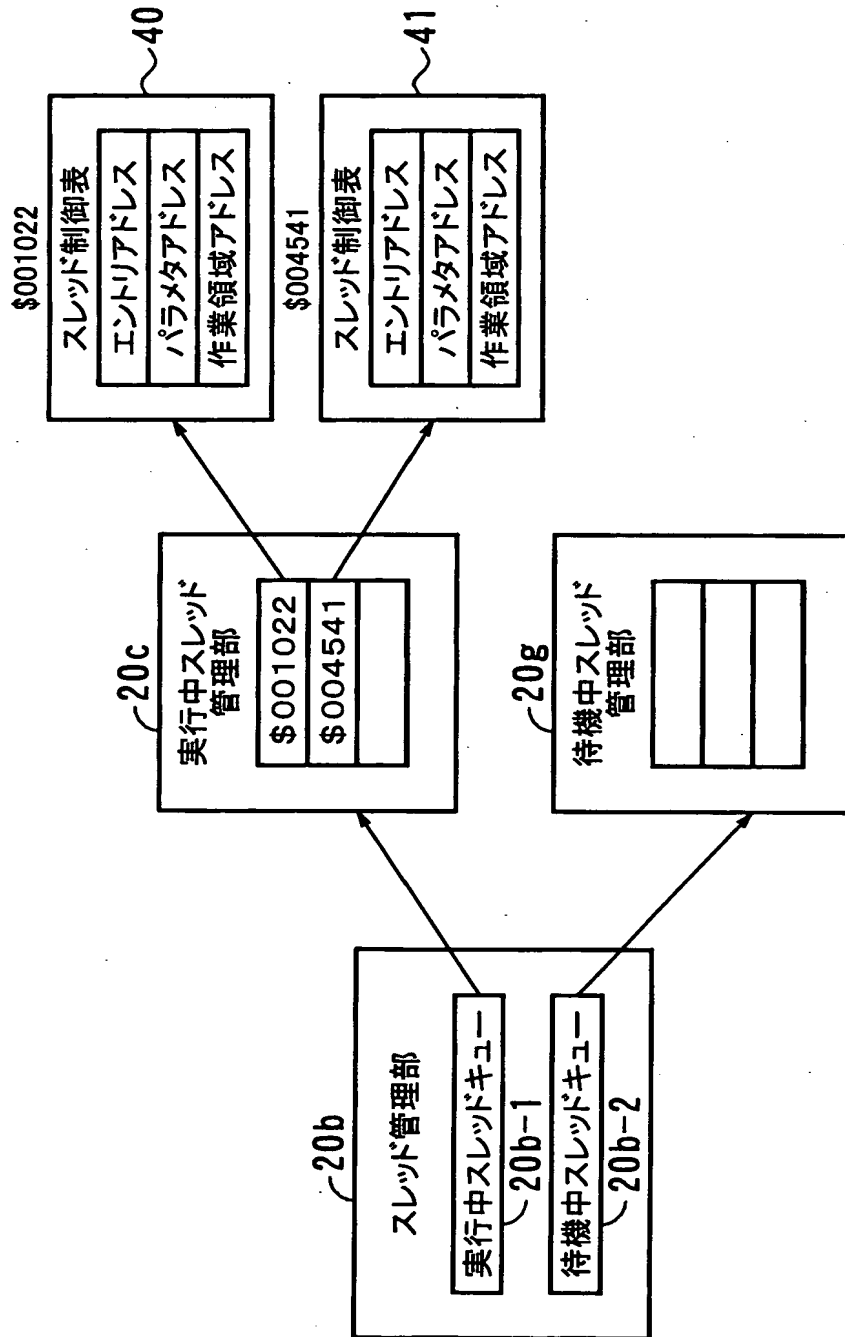
【図 2】



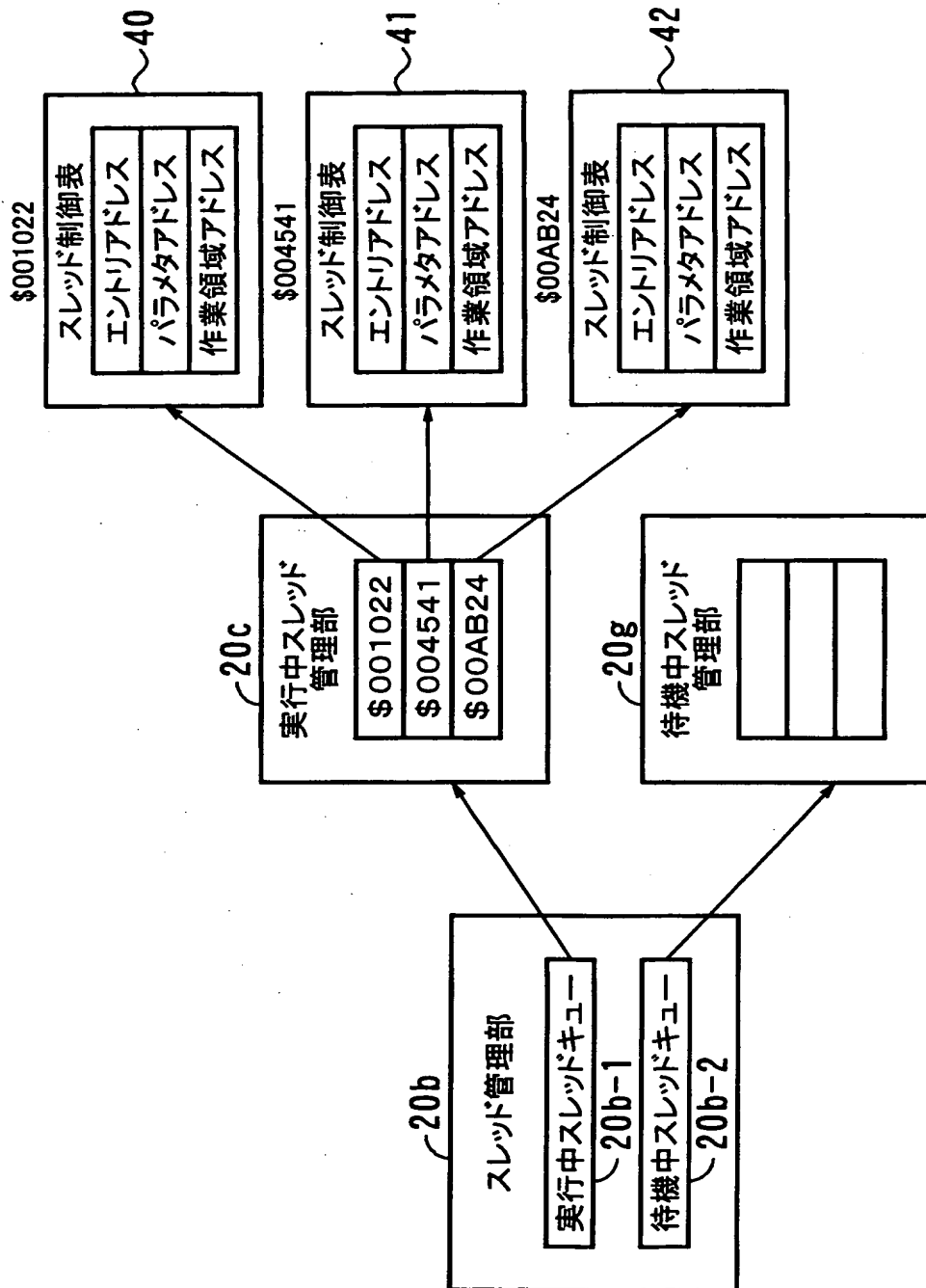
【図 3】



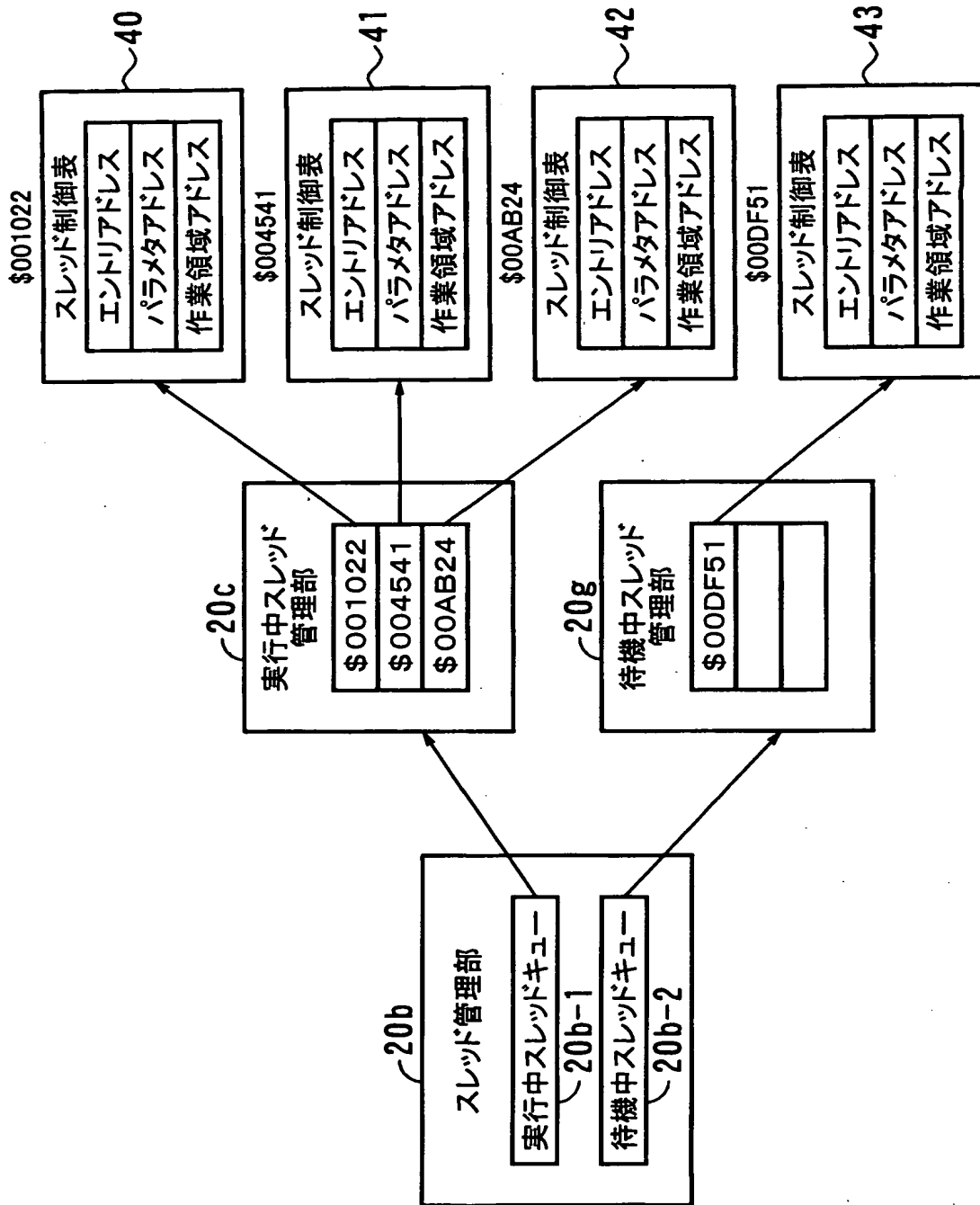
【図 4】



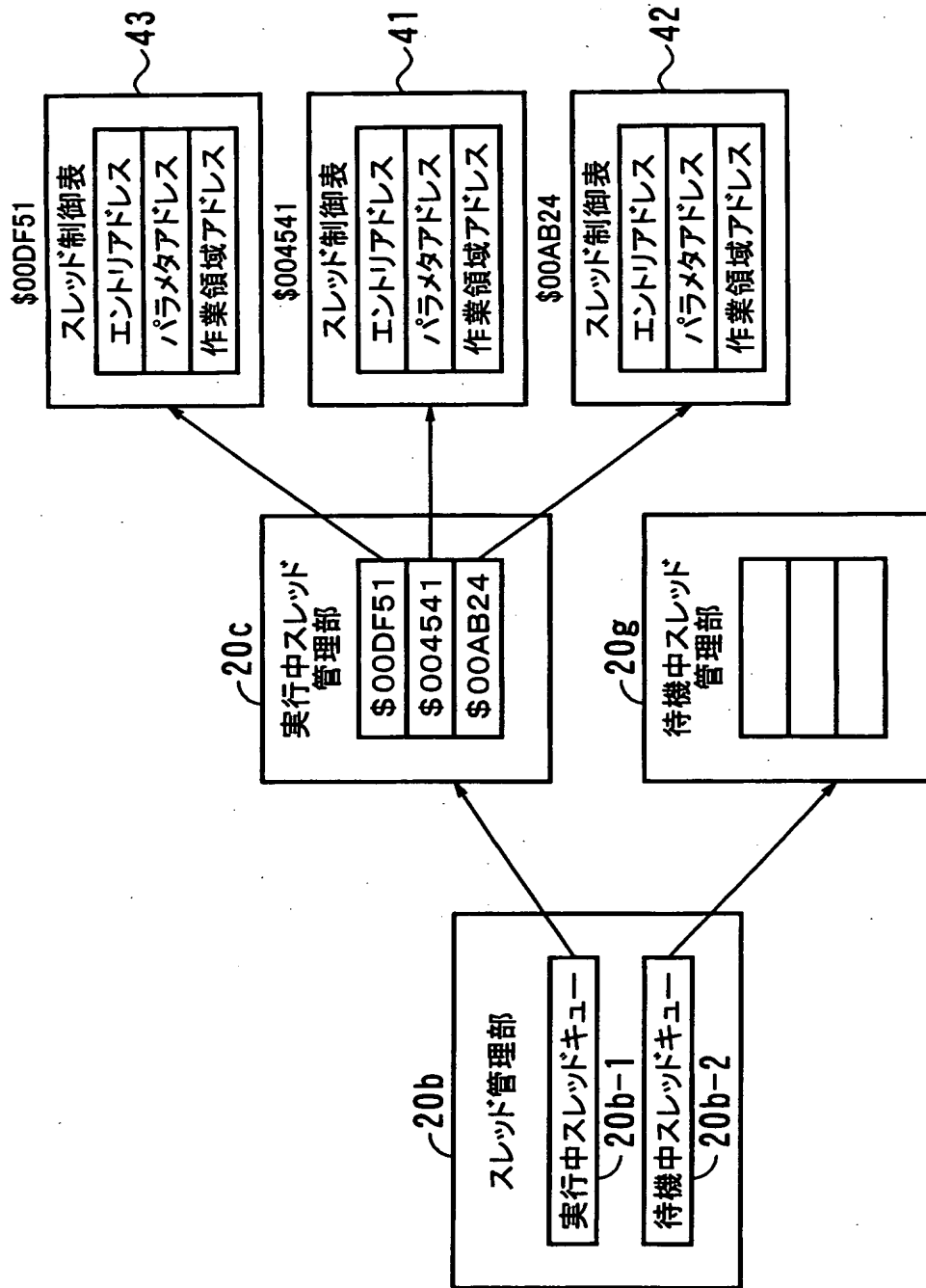
【図 5】



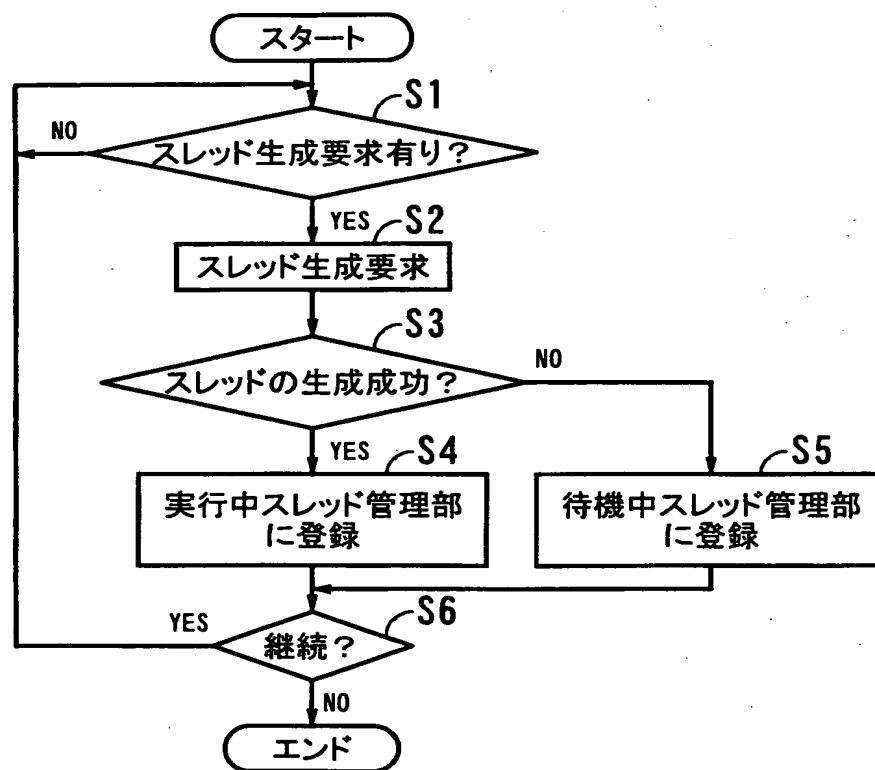
【図 6】



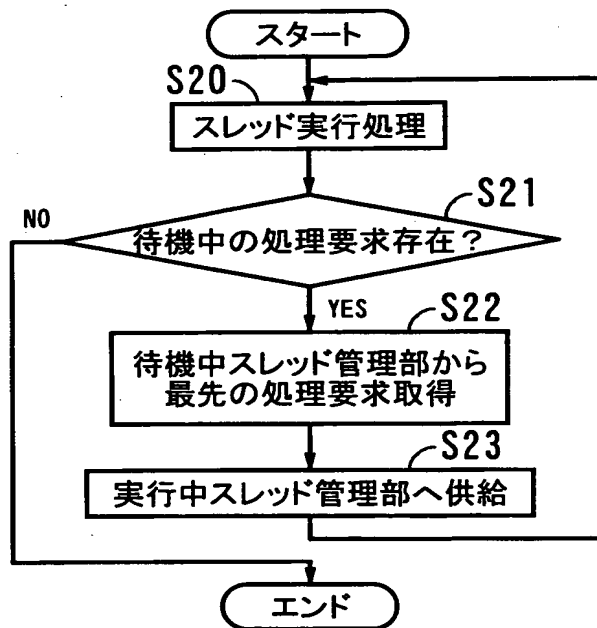
【図 7】



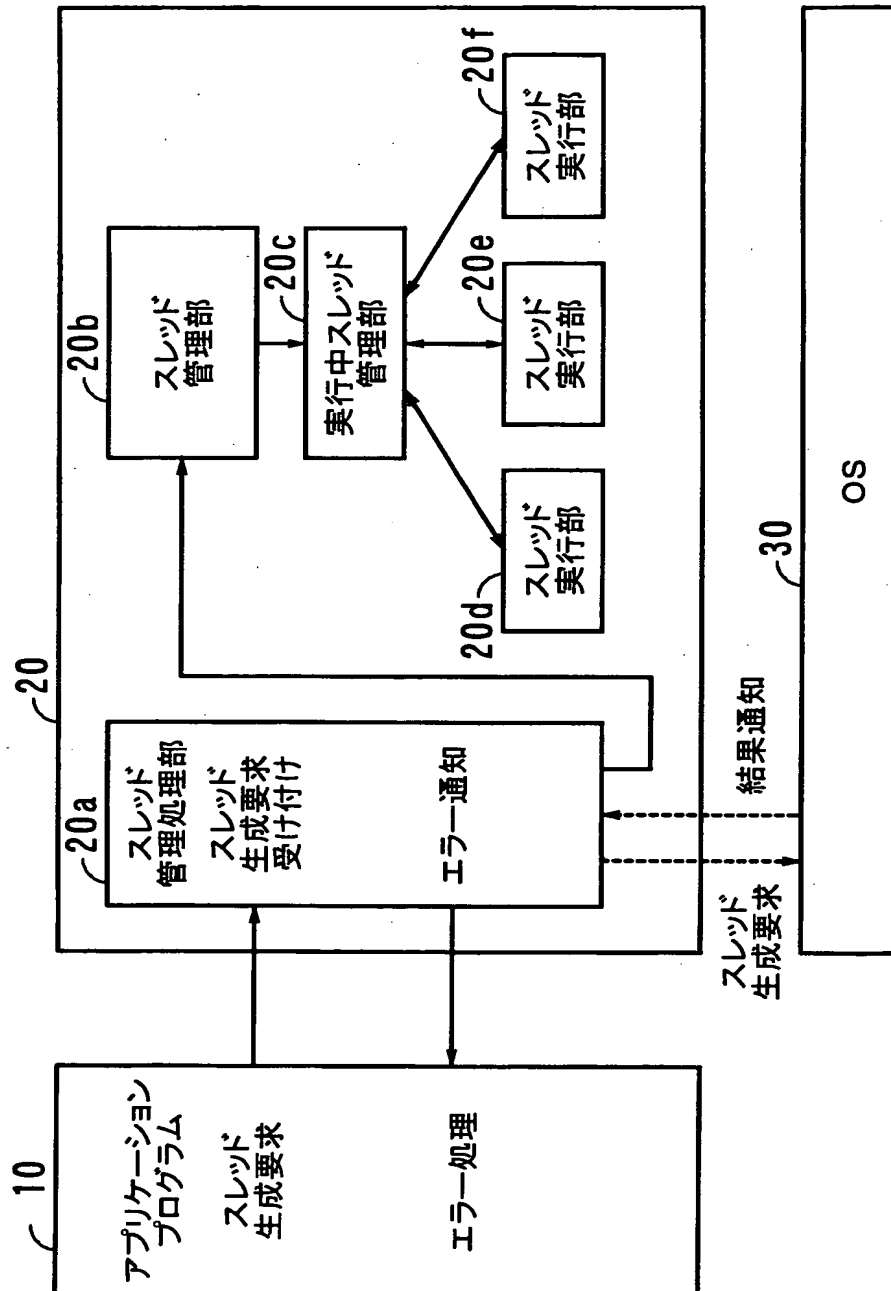
【図 8】



【図 9】



【図10】



【書類名】 要約書

【要約】

【課題】 マルチスレッドシステムにおいて、スレッドの生成に失敗した際の処理を迅速化する。

【解決手段】 スレッド生成要求受け付け手段 2 a は、アプリケーションプログラム 1 からのスレッド生成要求を受け付ける。スレッド生成手段 2 b は、スレッド生成要求受け付け手段 2 a によって受け付けられたスレッド生成要求に応じてスレッドを生成する。第 1 のスレッド実行手段 2 c は、スレッド生成手段 2 b によって生成されたスレッドを実行する。スレッド生成要求保持手段 2 d は、スレッド生成手段 2 b によるスレッドの生成に失敗したスレッド生成要求を保持する。第 2 のスレッド実行手段 2 e は、スレッド生成要求保持手段 2 d に保持されているスレッド生成要求に対応する処理を実行する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号
氏 名 富士通株式会社